

REMARKS

Applicant respectfully requests reconsideration and allowance of the subject application.

Claims 1, 16, 17, 24, 27, and 33-37 are amended.

Claims 1-17 and 20-37 remain pending.

Applicant thanks the Examiner for the detailed analysis presented in the current Office Action, just as the Examiner has provided detailed analyses in each of the preceding Office Actions.

CLAIM REJECTIONS UNDER 35 U.S.C. § 101

Claims 33-36 were rejected under 35 U.S.C. § 101 as not being directed to statutory subject matter. Claims 33-36, as well as claim 37, all have been amended to recite “a computer-implemented method.” Applicants submit that the amendments overcome the rejection under 35 U.S.C. § 101 to claims 33-36. Applicants respectfully request entry of the amendment and reconsideration of the claims.

CLAIM REJECTIONS UNDER 35 U.S.C. § 103

Claim 1 was rejected under 35 U.S.C. § 103 as being unpatentable over U.S. Patent No. 6,629,128 B1 to Glass (hereinafter "Glass") in view of U.S. Patent No. 6,058,391 to Gardner (hereinafter "Gardner"). Applicants have carefully considered the reasoning expressed in the preceding Office Actions. Applicants respectfully traverse the rejection, and submit that claim 1, as amended, is in condition for allowance.

1 The subject application is directed to challenges faced in managing systems
2 and devices in an enterprise environment. As computer systems and networks
3 continue to increase in size and complexity, so too does the challenge of managing
4 them. A significant tool that assists network developers and administrators in
5 managing computers across an enterprise is Windows[®] Management
6 Instrumentation (WMI). WMI enables the remote management of Windows-based
7 systems and applications by exposing management information through an object-
8 oriented structure defined using WMI schemas. WMI schemas are an
9 implementation of the Common Information Model (CIM) as defined by the
10 Desktop Management Task Force (DMTF).

11 WMI supports the management of systems and devices by exposing
12 management information across an enterprise, such as hardware settings,
13 performance information, driver configurations, BIOS information, application
14 settings, event log information, and so on, and by providing a mechanism to query
15 for information and configure settings on machines across the enterprise. WMI
16 provides access to management information on a single network machine, or a
17 large number of machines all at once. For example, without WMI, an
18 administrator wanting to enumerate descriptions for various groups of objects on a
19 machine must locate and learn different application programming interfaces
20 (APIs) that describe the specific methods for communicating with each group of
21 objects. However, WMI eliminates the need to learn the specifics of every API set
22 provided by Windows, through gathering information from a diverse set of APIs
23 and presenting this information in a simple, industry-standard management object
24 model.
25

Therefore, many comprehensive and well-documented managed resources are available to those developers and administrators capable of utilizing the benefits of WMI. However, WMI is generally designed for use by developers or administrators who are at least moderately proficient at programming in C/C++, Microsoft Visual Basic®, and scripts.

Writing such scripts, however, may be beyond the ability of many administrators, and the discovery of basic system information may therefore be difficult without the assistance of a more experienced programmer. In addition, much of the power of WMI is realized through developers writing management applications that monitor, configure and control the management information made available through WMI. Therefore, the benefits of WMI are often difficult to attain for the common administrator who does not have the proper programming background, but who still has a need to manage system components/objects.

Independent claim 1 has been amended, as have the other independent claims. Claim 1, as amended, is reproduced below:

1. (Currently Amended) A command line utility embodied in one or more computer-readable media, the command line utility comprising:

a command schema including one or more commands for engaging, from a management station on a network, an operating system on at least one selected target station accessible over the network enabling at least one of retrieval of management information from and initiation of a management service available through an object model target schema recognized by the at least one selected target station accessible over the a network;

1 an interactive user interface configured to receive the
2 one or more commands in the command schema from a user of the
3 management station, allowing the user to formulate an instant
4 request for at least one of management information and management
5 services and communicate a response of the at least one selected
6 target station to the user; and

7 an object model command schema to define a mapping
8 between the one or more commands in the command schema and the
9 object model target schema and interpret the one or more commands
10 from the command schema into a management command operable to
11 cause one of the retrieval of the management information from and
12 the initiation of the management service on the at least one selected
13 target station.

14 Applicants submit that claim 1, as amended, distinguishes over the cited
15 references.

16 Applicants submit that claim 1 is patentable over the cited references for at
17 least five reasons. First, Glass's a command line predevelopment utility for use by
18 developers does not teach or suggest the command line utility recited by claim 1.
19 Second, although Gardner describes a system that allows users to extract pre-
20 scripted views of a database without using SQL commands, it does not teach or
21 suggest, as recited by claim 1 as amended, an interface that allows a user to
22 formulate and enter instant commands. Third, neither Glass nor Gardner teaches
23 or suggests engaging an operating system on a target system. Fourth, neither
24 Glass nor Gardner teaches or suggests the operability of initiating a management
25 service on a target station as recited in claim 1. Fifth, neither Glass nor Gardner

1 teach or suggest that a user can use a management station for engaging an
2 operating system on a selected target station.

3 First, Glass's predevelopment command line fails to teach or suggest a
4 command line utility as recited by claim 1. Glass describes a system that
5 facilitates communications between two different programs by creating proxies
6 allowing a client application to communicate with a server-based object:

7 According to an embodiment of the present invention, a system for
8 distributed processing in a computer network is provided that
9 includes, a client side object request broker executing on a client
10 computer and a server-side object request broker executing on a
11 server computer. The server computer is connected to the client
12 computer through a network. *A remote proxy generator dynamically
13 generates remote proxy classes for client-side communications
14 support for communications between a client application and a
15 server object.* The remote proxy generator resides in the server-side
16 object request broker and instantiates the remote proxy class to
17 create a remote proxy object. A client-side type generator generates
18 a client side type object for a class of the server object. *The client-
19 side type object provides access to methods of the server object. A
20 client-side function generator generates one or more client-side
21 function objects for providing a connection to one or more
22 methods of the server object.*

23 (Glass, Column 3, Line 66, through Column 4, Line 16; emphasis added).

24 In this context, it is clear that Glass's description of an "interface" is an
25 interface between programs, not a user interface. The inter-object interface
contemplated by Glass is defined in the first paragraph of the background of the
invention:

Classes may also be characterized by their *interface which defines
the elements necessary for proper communication between objects.*

(Glass, Column 1, Lines 29-31; emphasis added). Furthermore, while there a few
mentions of users in the background of the invention of Glass (See Glass, Column
1, Line 46; Column 3, Lines 54-59), the word "user" is never mentioned in the

1 detailed description of the invention of Glass. Moreover, the phrase “user
2 interface” is never mentioned at all. Because Glass does not describe a user
3 interface, applicants respectfully submit that Glass neither teaches nor suggest the
4 elements recited in claim 1.

5 Respectfully, the Office Action is mistaken in its assertion that Glass’s
6 command line predevelopment utility discloses the command line utility recited by
7 claim 1. Specifically, the Office Action relies on Column 19, Lines 10-14 of
8 Glass, which describes what is, clearly, a *predevelopment utility*:

9 In one embodiment, interface generator 250 is a command
10 line *predevelopment utility used to generate interfaces* for classes
11 on server system that will be used remotely in distributed computing
12 system 100.

13 (*Glass*, Column 19, Lines 10-14; emphasis added). Moreover, this paragraph goes
14 to clarify for whom the command line predevelopment utility is directed, and how
15 it will be used:

16 In that embodiment, *the software developer* knows that
17 certain class files 252 will be used remotely. The *software
18 developer provides interface generator 250 with a list of class files
19 252 for which interfaces 254 are to be generated.*

20 (*Glass*, Column 19, Lines 14-20; emphasis added). Clearly, the only command
21 utility recited is the predevelopment utility by which a software developer can
22 create an interface, and nothing more.

23 By contrast, as recited in claim 1 as amended, the command line utility
24 includes “a command schema including one or more commands for engaging,
25 from a management station on a network, an operating system on at least one
selected target station accessible over the network enabling at least one of retrieval
of management information from and initiation of a management service available

through an object model target schema.” In other words, Glass is a predevelopment tool that describes a system that has a command line utility that allows for a software developer to create an interface that may allow control of a system. Claim 1, on the other hand, recites a command line utility “for engaging, from a management station on a network, an operating system on at least one selected target station.” Thus, Glass’s predevelopment tool for creating an interface does not render obvious claim 1’s command line utility that actually enables retrieval of management information from and initiation of a management service.

Second, even if for the sake of discussion one of ordinary skill in the art would have combined Glass with Gardner to overcome the shortcomings of Glass, applicants submit that Gardner fails to recite an interactive user interface as recited by claim 1. Gardner discloses a system for creating “customizable views of a database so that database users may view and update tables in a manner that requires no knowledge of the underlying database scheme or database commands.” (*Gardner*, Page 1, Abstract, Lines 1-4). However, like Glass, Gardner is a tool to be used by software developers or administrators, not by users:

A ‘view’ is created simply by having an *administrator* enter view information into the system. . . . The view information entered by an *administrator* includes the name of a table within the relational database that is to be associated with the view and the names of the columns from the table that are to be includes with the view.

(*Gardner*, Column 3, Line 66 through Column 4, Line 5; emphasis added).

Reading on, one sees that Gardner’s interface is all about what an administrator can do to create these views, and the users can use only the views created by administrators:

Database users are given access to the *stored views* through a view utility. To retrieve information from the relational database, the user simply uses the view utility *to select a view from among the views assigned to the user.*

(Gardner, Column 4, Lines 19-22; emphasis added). Thus, according to Gardner, users can only access views that already have been created, stored, and assigned to them. Nothing in Gardner allows users the freedom to control the command schema as recited by claim 1. Specifically, for example, claim 1 as amended recites “an interactive user interface configured to receive the one or more commands in the command schema from a user of the management station, allowing the user to formulate an instant request for at least one of management information and management services and communicate a response of the at least one selected target station to the user.” Gardner does not support user formulation of any command, let alone an instant request for management information or services. Gardner only allows users to engage previously created and assigned views, and thus does not afford the control given a user as expressly recited by claim 1.

Third, neither Glass nor Gardner teach or suggest a command schema for engaging from a management station on a network an operating system on a selected target station. As previously described, Glass facilitates distributed processing by between “client side objects” and “server side objects,” such that “The client-side type object provides access to methods of the server object. A client-side function generator generates one or more client-side function objects for providing a connection to one or more methods of the server object.” (Glass, Column 4, Lines 13-16). Neither Glass nor Gardner describes interaction with an

1 operating system as recited in claim 1. Thus, neither Glass nor Gardner teach or
2 suggest what is recited by claim 1.

3 Fourth, neither Glass nor Gardner recites the operability of initiating a
4 management service on a target station as recited in claim 1. The Office Action
5 concedes that Glass does not teach a command schema including one or more
6 commands enabling at least one of retrieval or management information and
7 initiation of a management service. Gardner fails to make up for this shortcoming.
8 As described, Gardner allows a user to access a view of a database, when an
9 administrator has created and stored that view and assigned it to the user. There is
10 no mention of a user being able to initiate a management service, or any other
11 service. Thus, neither of the cited references teach or suggest this element recited
12 by claim 1.

13 Fifth, neither Glass nor Gardner teach or suggest that a user can use a
14 management station for engaging an operating system on a selected target station
15 as recited by claim 1. As described at length previously, Glass and Gardner
16 describe predevelopment or administrator utilities that allow developers to create
17 interfaces or views for the users to permit them to engage applications.
18 Respectfully, however, neither reference allows a user to select a target station as
19 recited by claim 1. Users are constrained to access what developers and
20 administrators have allowed, wherever the resources needed to create the allotted
21 views reside, and do not give the user this type of control.

22 Thus, for these reasons, applicants submit that claim 1 as amended is
23 patentably distinguishable over the cited references. Applicants respectfully
24 request entry of the amendment and reconsideration of claim 1.
25

1 Claims 2-15 depend from and add additional limitations to claim 1 and,
2 thus are patentable for at least the same reasons as claim 1.

3 Notwithstanding, with regard to the objections to claims 2-15, applicants
4 note that claim 1 as amended, from which these claims depend, describes how
5 management information can be retrieved from and management services initiated
6 on a selected target station. In rejecting claims 2-15, the Office Action combines
7 U.S. Patent No. 6,560,591 to Memmott et al. (hereinafter, "Memmott") to
8 conclude that the claims are not patentable. However, even if for the sake of
9 discussion one of ordinary skill were to combine Memmott with the other
10 references, Memmott actually teaches away from the invention recited by claim 1.
11 Specifically, Memmott describes a system whose very object is that the user does
12 not have to be aware of the target system: "It is desirable to free a requesting entity
13 from the burden of having to select a particular provider." (*Memmott*, Column 3,
14 Lines 7-8). Accordingly, because Memmott focuses on device independence, it
15 teaches away from what is recited by claim 1.

16 Thus, for the reasons claim 1 is patentable, and for at least these additional
17 reasons, applicants submit that claims 1-15 are patentable over the cited
18 references. Applicants respectfully request entry of the amendment and
19 reconsideration of the claims.

20 Other independent claims, including claims 16, 17, 24, and 33, also are
21 rejected under the combination of Glass, Gardner, and Memmott, based on the
22 Office Action's reincorporation of the bases listed in the Office Action with regard
23 to other claims. Applicants respectfully reincorporate its remarks with regard to
24 claim 1 with regard to these other claims. For the reasons described previously,
25 claims 16, 17, 24, and 33 as amended, and the claims depending therefrom,

1 including claims 20-23, 25-26 and 34-37, are patentable over the cited references.
2 Applicants respectfully request entry of the amendments to claims 16, 17, and 33,
3 and reconsideration of claims 16, 17, 20-26, and 33-37.

4 The Office Action rejects claims 27-32 over Memmott in view of Gardner
5 in further view of publication "Network and System Management with XML" by
6 Steve Steinke (hereinafter, "Steinke"). Respectfully, claim 27 as amended is
7 distinguishable over the references to Memmott and Gardner for the reasons
8 already described, and Steinke does not make up for the shortcomings of the other
9 references.

10 In addition to the reasons already stated, applicants wish to emphasize that
11 claim 27, as amended, recites "receiving a command from a user that was
12 formulated as an instant request by the user at a management station through an
13 interactive command line interface," and "executing the command as one or more
14 WMI API calls against a targeted namespace representing at least one selected
15 target system that is accessible via a network." As previously described, Gardner
16 describes a system that allows an administrator to create views for a user, and that
17 is the only access users have. Accordingly, Gardner does not permit a user to
18 formulate an instant request at a management station as recited by claim 1.
19 Similarly, because its very object is device and destination transparency, Memmott
20 does not provide the user with the capability of executing a command against a
21 selected target system. Steinke does not make up for these shortcomings. Thus,
22 for these reasons, applicants respectfully submit that claim 27 is patentable over
23 the cited references. Applicants respectfully request entry of the amendment and
24 reconsideration of claim 27.
25

1 Claims 28-32 depend from and add additional limitations to claim 27 as
2 amended. Thus, applicants respectfully submit that claims 28-32 also are
3 patentable over the references cited, and request reconsideration of these claims.
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25

CONCLUSION

Claims 1-17 and 20-37 are in condition for allowance. Applicant respectfully requests reconsideration and prompt allowance of the subject application. If any issue remains unresolved that would prevent allowance of this case, the Examiner is requested to urgently contact the undersigned attorney to resolve the issue.

Respectfully Submitted,

Date: 5-1-2006

By:



Frank J. Bozzo
Lee & Hayes, PLLC
Reg. No. 36,756
(206) 315-4001 x103 or
(206) 315-7918